

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **59062968 A**(43) Date of publication of application: **10.04.84**

(51) Int. Cl.

G06F 15/16
G06F 3/00
G06F 11/00

(21) Application number: **57170841**(22) Date of filing: **01.10.82**(71) Applicant: **HITACHI LTD**

(72) Inventor:
OYASHI MASANAO
FUSHIMI HITOSHI
NAKANISHI HIROAKI
YASUMOTO SEIICHI
OKADA MASAKAZU
SUEKI MASAO
HAYASHI KEIJIROU
ONUKI TAKESHI
IDE TOSHIYUKI
MIZOKAWA SADA0

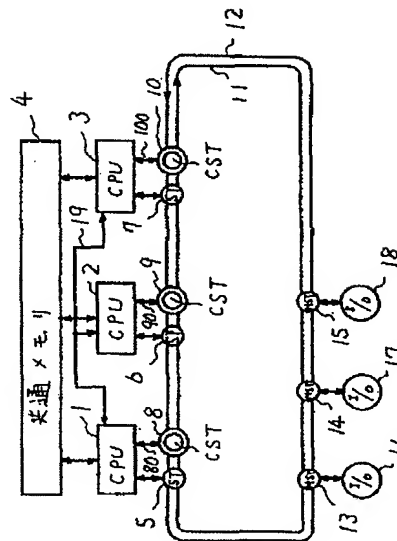
(54) **CONSTITUTION CONTROLLING SYSTEM OF
 CONTROL STATION IN MULTICOMPUTER
 SYSTEM**

COPYRIGHT: (C)1984,JPO&Japio

(57) Abstract:

PURPOSE: To prevent the inconvenience that common busses cannot be managed, by allowing a CPU to manage states of all control stations collectively with a common memory to perform the constitution control of each control station.

CONSTITUTION: When a CPU1 which is connected to a master control station CST8 receives the fault report of the CST8 itself from the master CST8, the CPU1 refers to constitution control information in a common memory 4 to recognize the CST8 connected to the CPU1 as the master CST and rewrites this master CST with an idle CST and reports it to a CPU2 that the CST 8 is faulty. The CPU2 refers to constitution control information in the common memory 4 and issues a command to use a CST9 connected to the CPU2 as the master CST if the CST9 is a monitor CST. In this case, the constitution control is performed surely because the common memory 4 cannot be accessed simultaneously by two CPUs.



⑬ 日本国特許庁 (JP)

⑪ 特許出願公開

⑫ 公開特許公報 (A)

昭59—62968

⑤ Int. Cl.³G 06 F 15/16
3/00
11/00

識別記号

庁内整理番号

L 6619—5B
Z 6549—5B
7368—5B

④ 公開 昭和59年(1984)4月10日

発明の数 1

審査請求 未請求

(全 12 頁)

⑤ マルチ計算機システムにおけるコントロール
ステーション構成制御方式

② 特 願 昭57—170841

② 出 願 昭57(1982)10月1日

⑦ 発 明 者 大林正直

日立市大みか町5丁目2番1号
株式会社日立製作所大みか工場
内

⑦ 発 明 者 伏見仁志

日立市大みか町5丁目2番1号
株式会社日立製作所大みか工場
内

⑦ 発 明 者 中西宏明

日立市大みか町5丁目2番1号
株式会社日立製作所大みか工場
内

⑦ 発 明 者 安元精一

日立市大みか町5丁目2番1号
株式会社日立製作所大みか工場
内

① 出 願 人 株式会社日立製作所

東京都千代田区丸の内1丁目5
番1号

④ 代 理 人 弁理士 高橋明夫

最終頁に続く

明 細 書

発明の名称 マルチ計算機システムにおけるコ
ントロールステーション構成制御
方式

特許請求の範囲

1. 複数の計算機と、該複数の計算機に接続され
計算機間で共通に使用される共通メモリと、該複
数の計算機を接続する共通バスと、該共通バスに
接続され、該計算機によつて制御される複数の入
出力機器と、該共通バスに接続されるものであつ
て、同時には1つのみがマスタステーションと
して共通バスの統括管理を行うように制御される
複数のコントロールステーションを有するマルチ
計算機システムにおいて、各計算機と複数のコン
trolステーションを対応づけて接続し、各コ
ントロールステーションは共通バスが正常に使用
されているか否かの状態信号を対応する計算機に
報告し、各計算機は、対応するコントロールステ
ーションからの該状態信号および他計算機からの
指令にもとづき、該共通メモリに格納された構成

制御情報を参照し、該構成制御情報の更新と、対
応するコントロールステーションへの構成制御指
令を出すようにしたことを特徴とするマルチ計算
機システムにおけるコントロールステーション構
成制御方式。

発明の詳細な説明

〔発明の利用分野〕

本発明は複数の計算機が、共通バスに接続され
た複数の入出力機器を制御するマルチ計算機シ
ステムに関し、特に共通バスを統括管理する制御機
能を有する複数のコントロールステーションの構成制御
方式に関するものである。

ここで、「構成制御」とは、共通バスに接続さ
れた複数のコントロールステーションのうち、常
時、ただ1つがマスタステーションとして共通
バスの統括管理を行うようにし、マスタステ
ーションが故障した場合、ただちに、残りの待機中
のコントロールステーション(モニタステ
ーション)の1つが今までのマスタステーションに
代わり、共通バスの統括管理を実行するようにコ

ントロールステーションを制御させることを意味する。

ここで、「統括管理」とは、共通バスの使用に関する全ての制御或いは管理であり、計算機が共通バスを占有し、特定の入出力機器を制御するようなシステムにおいては、共通バスの占有制御が大きな仕事である。

以下、統括管理をするコントロールステーションをマスターCST、マスターCSTが故障時に、これに代つてマスターCSTになりうるコントロールステーションをモニターCST、これ以外のコントロールステーションをアイドルCSTと称す。
〔従来技術〕

共通バスの一形態はループ状伝送路である。

近年、ループ状伝送路を多重化し、伝送路そのものの信頼性を向上させたものが提案されている。また、ループ状伝送路には、複数のCSTが接続され、任意の1つをマスターCST、残りをモニターCST或いはアイドルCSTとし、CSTの構成制御により、特定のCSTがダウンした場合

でも、ループ状伝送路に接続されたシステムが、システムダウンにならないようにしている。

従来、CSTは計算機と独立に、共通バスに接続され、各CSTは、独自に、マスターCST、モニターCSTを決定するようになっていた。

すなわち、モニターCSTは、共通バス上の信号を監視することにより、マスターCSTが故障したか否かを判断し、故障したと判断した場合、自CSTをマスターCSTとし、共通バスの統括管理を行うようにしている。この場合、モニターCSTが2台以上ある場合、どちらのモニターCSTがマスターCSTになるかは、予め決めている場合が多いが、これの管理は複雑で、同時に、2つのモニターCSTがマスターCSTになろうとする場合がある。

また、マスターCSTそのものに故障がない場合であつても、モニターCSTは、共通バス上の信号が途絶えた場合は、マスターCSTの故障とみなして、自分がマスターCSTになろうとする場合がある。

このようなことがあつて、従来は、CST間に、特別な信号線を設け、信号のやりとりにより、このような不都合を防止しようとする試みがあるが制御が複雑となり、確実なCSTの構成制御は困難であつた。

〔発明の目的〕

本発明の目的は、共通バス上の複数のCSTのうち、1個のみをマスターCSTとし、その他のCSTを、モニターCST又はアイドルCSTとなるように確実に制御するマルチ計算機システムにおけるコントロールステーション構成制御方式を提供するにある。

〔発明の概要〕

本発明の特徴は、共通バスに接続された入出力機器を制御する複数の計算機と各CSTを対応づけて接続し、モニターCSTをマスターCSTにするのは対応する計算機からの指令によつて行うこと、各CSTは当該CST自身の状態および共通バス上の信号の状態を対応する計算機に報告すること、および、どのCSTがマスターCSTに

なつているかの情報は、各計算機が共通に使用する共通メモリに格納し更新しておき、各計算機は、対応するCSTからの報告及び、他の計算機からの指令により、共通メモリの構成制御情報を参照して、構成制御を行うことである。

本発明によれば、構成制御の情報は、1ヶ所の共通メモリ上で管理されるので、誤つて2つ以上のCSTがマスターCSTとなることはない。

マスターCSTに接続されている計算機は、マスターCSTから、CST自身の故障報告を受けると、共通メモリの構成制御情報を参照し、自分に接続されているCSTがマスターCSTになっていることを認識し、ただちにこれをアイドルCSTに書き換え、他の計算機へ、当該CSTがダウンしたことを報告する。

この報告を受けた計算機は共通メモリの構成制御情報を参照し、自分に接続されているCSTがモニターCSTの場合、これをマスターCSTに書き換え、自分に接続されているCSTに対し、マスターCSTとなるよう指令を出す。

この場合、共通メモリは、同時に2つの計算機からアクセスできないので、確実な構成制御ができる。

〔発明の実施例〕

第1図は本発明が適用されるマルチ計算機システムの一実施例である。

第1図において、1～3は、計算機（以下CPUと称す）であり、プログラムを内蔵し、このプログラムを実行することにより、複数の入出力機器16～18を制御するものである。4は、各CPU1～3が共通に使用する共通メモリであり、後述するように、この中にCSTの構成制御情報が格納される。5～7は各CPU1～3を共通バス、ここでは逆向きのループ状伝送路11、12に接続するステーション（以下STと称す）であり、8～10は、ループ状伝送路11、12の統括制御機能を持つコントロールステーション（CST）である。CST8～10はそれぞれ接続線80、90、100を介して対応するCPU1、2、3に接続されている。13～15は入出力機器（以

下I/Oと称す）16～18をループ状伝送路11、12に接続するマイクロステーション（以下MSTと称す）である。19は、各CPU1、2、3間を接続する共通信号線で後述するように、CSTの構成制御時に用いられる。各MSTには自身の下に接続されているI/OがどのCPUからの要求に対してサービスをすべきかを記憶する構成及びCSTからの指令に基づいてループを折り返す機能を有している。どのCPUからの要求に対してサービスをすべきかは、サービスすべきCPUが確定していない状態（この状態をニュートラル状態と称す）で、CPUからの占有要求を該MSTが受取つたときに記憶され、この占有状態は該占有中のCPUから占有解除の指令を受けたとき、および他の占有中でないCPUから強制的に占有解除の指令を受けたときに解除される。この占有要求指令をCONC・RSV（Connect & Reserve）コマンド、占有状態の解除指令をCONC・FREE（Connect & Free）コマンド、強制的に占有を解除する指令をRESET・FREE

（Reset & Free）コマンドと称す。MSTは自分の下に接続されているI/Oに対する指令がCPUから送られてきたとき、そのI/Oを占有しているCPU以外からの指令であれば、それがRESET・FREEコマンドでないかぎり、その指令を受け付けない。この機構を設けることによつて、ループバスにMSTを介して接続されているI/Oは、複数CPUから適正に共有される。即ち、CPU1から占有されているI/O16に対して、CPU2のユーザプログラムから誤つて入出力要求が発行されてもI/O16は現にサービス中の処理に乱れを生じない。CPU2からの入出力要求はMSTによつて拒否されるからである。

各CPUは、このようなI/Oの接続状態の管理を行つており、各CPU内のメモリには、このI/Oの接続状態を管理するテーブルが置かれている。各CPUのオペレーティングシステム

（OS）は、I/Oの占有状態を切り換える必要が生じたとき、現に該I/Oを占有しているCPUに連絡をし、該I/Oに対してCONC・FREEコ

マンドを発行するよう依頼する。依頼を受けたCPUはCONC・FREEコマンドを該I/Oに発行し、占有を解除し、依頼元CPUに対して占有を解除した由の応答を返す。依頼元CPUはこの応答を受け取つてからCONC・RSVコマンドを発行し、該I/Oを占有する。現に該I/Oを占有しているCPUが停止状態ないし正常に動作していないことが検出されたときは、依頼元CPUは、RESET・FREEコマンドを発行し該I/Oの占有状態を解除後、CONC・RSVコマンドを発行し該I/Oを占有する。これらのI/Oの占有の遷移に従い、各CPU内の管理テーブルを書き換えるが、この管理テーブルは、CPU間共有メモリ4に置いてよい。

系の信頼性を高めるためにCSTは同一ループに複数接続される。CSTが1つであるとそのCSTが故障する場合があることもさることながら、そのCSTの両側のステーションで異常が生じたとき、CSTから他のステーションへ制御情報が伝えられなくなり、ループが機能しなくなる

からである。この複数CSTが互いに独立ループを制御するようにすると、相互に矛盾した制御を行なうことがあり得るので、複数CSTのうち1つだけにループを制御する権限を与える。この権限を与えられたCSTをMasterモードのCST(マスターCST)と称している。他のCSTは、ループの状態監視をするモード(Monitorモード)にあるか、又は単に受動的に動作するモード(Idleモード)のいずれかにあり、MonitorモードにあるCSTをモニターCSTと称している。マスターCSTからのみMST, STに対してループを折り返すよう指令できるとともに、通常CPUとI/Oとのデータ伝送に使われるループ11と、ループの監視信号等を伝送するために使われるループ12の切り替え等の指令ができる。マスターCSTはループに対して監視信号を送出する。モニターCSTは、マスターCSTから送られてくる監視信号を監視し、この監視信号が一定期間検出できなかつたとき、マスターCSTが故障したと判断し、CPUに対して割り込みをか

極的に働きかけをすることはせず、またループの状態の監視も行なわない。アイドルCSTないしモニターCSTは、CPUからマスターになるよう指令(この指令をマスターコマンドと呼ぶ)を受けるとマスターとなり、ループの制御権を得る。また、マスターCSTは、CPUからのリセット指令(リセットコマンド)によりモニターCSTに移行する。この状態の遷移を第2図に示している。CSTは電源をONされると、アイドルモードとなり、CPUからCONC・RSVコマンドを受け取るとモニターモードに移行する。これは、システムを最初にスタートさせるとき、電源投入シーケンスの違いにより無用なエラー処理をしなくてすむようにするためである。即ち、CSTの電源が最初に投入されCSTをモニターモードとすると当該モニターCSTはループの監視を開始する。このタイミングとマスターモードCSTが定義されるまでの間(マスターモードのCSTはCPUからの指令によつて定義されるので、CPUの電源が投入され、動作可能となる以前にはマス

ターモードのCSTが存在しない期間が生じる)にループが異常である(マスターモードCSTの異常)とモニターモードのCSTがこれを判断してしまうことになるが、CPUからの指令によりモニターモードに移行することとすればこの問題は生じない。

ループバスの運転モードの遷移を第3図に示している。NL運転とは通常の運転形態であり2重化ループの一方(これをNL: Normal Loopと呼ぶ)でCPUとI/O間のデータ伝送を行ない、他方(これをBLと呼ぶ)でループの監視を行なうものである。NL運転中に、Normal Loopに異常が検出されると、Back LoopにてCPUとI/O間のデータ伝送が行なわれる。Normal Loop, Back Line共に異常(たとえばループが2本とも断線したとき)となつて、ループ一巡信号が検出されなくなると、マスターCSTは、各ステーションに対してループの折り返しを指示し、ループを一巡して(Normal Loopと折り返し点とBack Loopと折り返し点を結ぶループ)

有意な信号が伝送し得るようループの構成制御（ループバック制御）を指示する。これによりループ一巡する有意な信号が存在し、ループバスとして機能をはたすようになっていく運転モードをLB運転（Loop Back 運転）と称し、折り返し点には含まれたステーションはループから切り離された状態となる。第3図で破線矢印は、Master モードにあるCSTに対してCPUから通常運転への復帰コマンド（RLBCコマンド）が発行されたときに運転モードが遷移することを示す。CPUからMaster モードのCSTに対してLB運転に移行するよう指令（SLBCコマンド）することができる。このときMaster モードのCSTには折り返し点となるべきステーションを指示するが、Master モードのCSTは記憶しているループの構成情報に従って処理を行なう。

以上の動作を具体的に図面を用いて説明する。

ここで、CPU1及び、CPU1接続CST8を例にとつて説明する。CPU2, 3及びCST9, 10も同様である。

Loop CONTROL 8-14、信号受信回路8-20, 21、信号送信回路8-18, 19よりなる。又、接続計算機への報告用として、報告レジスタ8-27がある。割込みレジスタ8-9および報告レジスタ8-27は接続線80を介してCPU1に接続されている。以上のような構成を用いて、具体的な処理を、第5図から第9図を用いて説明する。第5図は、CST8内の、CPU1からの指令又は、CST8の電源ONにより起動される処理を示す。まずCST8は、電源ON時は、無条件にIDLEモードへ移行するとともにIDLEモードをRAM8-3へ記憶する。IDLEモードでは、自らはLoop 11, 12に対し、Pass 状態へ移行する。これはLoop CTL8-14及びマルチプレクサ8-17又は8-22により行なわれる。又CPU1からの指令が、MASTER指令であつた場合、自らはMASTERモードである事をRAM8-3へ記憶し、Loop 11（Normal Loop）へ監視信号を送出するとともに監視信号一巡待ちタイマ8-7を起動する。又、

第4図にCST8のブロック図を示す。CST8は、マイクロプロセッサ8-1、マイクロプログラムを内蔵する脱出し専用メモリ（ROM）8-2、書き込み可能なメモリ（RAM）8-3、割込制御LSI（PICU）8-4、通信制御LSI（HDLC）8-5、汎用入出力制御LSI（PPI）8-6、タイマ（PTM）8-7、およびCPU1からの割込の内容を記憶する割込レジスタ8-9からなりこれらは共通BUS8-8にて接続される。

更に、データ伝送を促す、監視信号発生器8-11からの監視信号とHDLC8-5からのデータ信号のどちらかを選択するマルチプレクサ（MPXB）8-12、マルチプレクサ（MPXB）8-12により選択された信号を送信するか、ループ上の信号をそのまま送信する（IDLE）かを選択するマルチプレクサ（MPXA）8-17、および逆方向のマルチプレクサ（MPXD）8-22、複数ループ全ての信号断を検出する検出器8-15、およびこれらマルチプレクサへ選択指令を発する

CPU1からの指令がリセット（Reset）であつた場合、自らは、MONITOR モードへ移行し、Loop 11, 12に対してはPASS状態へ移行する。更に、相手MASTER CSTからの監視信号待ちタイマ8-7を起動する。第6図は現用ループからの割込により起動される割込み処理の一例を示す。ここで現用とは、Normal Loop 運転中はLoop 11、Back Loop 運転中はLoop 12の事を言う。割込みは、Loop 11又は12からの受信はマルチプレクサ8-13を介してHDLC8-5が受信し、HDLC8-5からの割込が割込制御LSI8-4にてMPU8-1へ受信割込みが入る。この割込により第6図に示す割込処理プログラムが起動される。この時、受信データが、監視信号であつた場合、MONITOR CSTは監視信号待ちタイマ8-7を再起動させる。又、この割込が、HDLC LSI故障の時は、自CSTがMASTERの場合、Loop 11, 12の管理が維持できないと判断し、Loop に対しPass 状態へ移行し、IDLEモードへ移行する。更に、接続

CPU1へ自CST8 MASTER 故障を報告レジスタ8-27へセットし報告する。

第7図は、現用ループ11又は12の信号断のチェック処理の一例である。信号断のチェックはMASTER CSTのみにて行い、信号断は信号断検出器8-15より検出し、汎用入出力制御LSIのステータスへ反映させる事により検出する。信号断がNormal 運転中 (Loop 11)であつた場合、現用ループをBack Loop (Loop 12)へ切換える。切換処理は、Loop CTL 8-14の指令によりマルチプレクサA 8-17をしや断し、マルチプレクサD 8-22に対しマルチプレクサB 8-12からの信号をLoop 12へ送出するようになる。更にBack Loop (BL) 12に対し、監視信号を送出すると共に、Normal Loop 故障をCPU1へ報告する。信号断の時、Back Loop 運転中であつた場合、Loop 11, 12へ接続されているST5, 6, 7およびRST13, 14, 15に対し、Loop Back 指令を送出し、Loop Back モードへ移行する。更にBack Loop 異常

セス)にて行なわれた場合、DMAエラーとなつた時、自CSTと接続CPUとの交信手段が途絶えたとしてRAM 8-3内CSTモードがMASTER モードであつた場合、IDLEモードへ移行し、Loop に対してはPASS状態とする。

ループに接続されたI/O (16~18)はそれぞれが別のCPUに占有される事が可能であるが、ループに接続されている全I/Oを一括してあるCPUに占有させることも行なえる。これを行なうためにはループを一括して占有しているCPUから該ループ下の全I/Oに対してCONC・FREE コマンドを発行し、新たにループを占有したいCPUから該ループの全I/Oに対してCONC・RSV コマンドを発行する。このときループがLB 運転状態であり、I/Oがループから切り離されていた状態であると該I/Oに対してはCONC・RSV コマンドは届かないので、該I/Oのみ新たにループを占有したCPUには占有されないことになる。この不都合を防ぐため、CPUは、Master モードにあるCSTからの判り込み

をCPU1へ報告する。又、信号断が、Loop Back 運転中であつた場合、Loop 11, 12を管理できないと判断し、接続CPU1へ、自MASTER CST Give up を報告する。第8図はCST内のタイムアウト処理の一例を示すもので、RAM 8-3記憶のCSTのモードがMONITOR モードであつた時に、相手MASTER モードCSTからのPOL 信号を前もつて定め一定時間内に受信しなかつた場合、本タイムアウトとなる。本タイムアウトが発生した場合、相手MASTER モードCSTの故障と判断し、CPU1へ判定により報告する。又RAM 8-3記憶のCSTモードがMASTER モードであつた場合、自CST8が送出した監視信号がある一定時間内に監視信号一巡待タイムアウトとなるが、本タイムアウトとなつた場合、自MASTER CST 故障をCPU1へ報告しIDLEモードへ移行する。第9図は、自CSTとCPUとの交信手段が途絶えた場合の処理の一例である。本例の場合、CPU1への報告がDMA (ダイレクト・メモリ・アク

により報告される切り離されたI/Oがどれであつたかを記憶しておき、ループを通常運転に戻したあと (Master モードのCSTに対してRLBC コマンドを発行したあと) 該I/OにCONC・RSV コマンドを発行する。このようにすることにより、ループの運転状態によらず、各CPU間でループを一括占有、占有の切り換えが可能となる。

さて、マルチコンピュータシステムにおいては、CPUの構成制御と云うことが行なわれる。たとえば、第1図でCPU1, 2, 3で業務A, B, Cをそれぞれ実行しているとして、業務Aがシステム全体にとって重要な仕事であつたすると、CPU1が異常を生じてダウンした時には残りのCPU2, 3で業務B, Cを継続してもシステム全体として意味がないことがある。従つてこのような場合には、CPU2ないし3によつてCPU1で行なわれていた業務Aを引き継がねばならない。また異常を生じたCPU1が占有していたリソースを全て解放して、他のCPU2, 3が処理

を行なう上で、障害、外乱を与えないようCPU 1を停止せしめてやらねばならない。このような処理をCPUの構成制御という。また、CPUの異常がなくとも、システム運転の都合上、計画的にあるCPUの動作を停止せしめること、CPU内で行なわれている業務内容の切り換えを行なうこともある。これもCPUの構成制御と呼ぶ。このような構成制御を行なうとき、MasterモードのCSTが接続されているCPUが結果として停止するような構成制御を行なうときには、他の生き残るCPUに接続されているCPUに接続されているCSTをMasterモードとしなければならない。なぜならば、接続されているCPUが停止している状態では、MasterモードのCSTはループの異常を報告することができないし記憶しているループの構成情報をCPUからの指示により更新することもできないからである。

また、MasterモードのCSTから該CSTがMasterモードを維持できなくなつた由の連絡を受けたCPUは、他のCPUの1つに対して、接

続されているCSTをMasterモードとするよう連絡を出す。

CPUの構成制御に伴なうMasterモードのCSTの切り換えは第10図のように行なう。第10図で「CPUの切り離し」とは、対象CPUを停止せしめ、該CPUが占有していたリソースを解放する処理を言う。「CPUの切り換え」とは、そのCPUでそれまで行なわれていた業務を停止せしめ、該業務が占有していたリソースを解放せしめ、新たな業務を開始する処理を言う。

「CPUの立上げ」とは、それまで停止していたCPUを動作開始させ、新たな業務を開始する処理を言う。CPUの切り離し、立上げの場合の処理フローを、第11図、第12図にそれぞれ記す。第12図において、ループがCPUに接続されているか否かは、たとえば第13図のごときテーブルをもつことによつて調べられる。第13図で、31は各CPUから共通に参照できるCPU間共有メモリである共通メモリGM(第1図の4)に格納しておく。このテーブルはシステム全体のル

ープバスの数だけ(ループ1からループnまで)あり各ビット0, 1, 2...をCPUの番号に対応させておく、即ち、ループ2用のビット2が1ならばループ2はCPU # 2のCPUに接続されており、0ならばループ2はCPU # 2のCPUには接続されていない事を示す。また、このテーブルは他のCPUの下でCSTをMasterモードとするよう依頼するときにも用いられる。ループに接続されているCSTのうちどれがMasterモードにあるかを求めるのにも第13図と同一の構造のテーブルを用いる。但し、MasterモードCSTを管理する場合、MasterモードのCSTを“1”で表わすと第13図において、各ループ用のテーブルで“1”であるビットは各列に唯1つだけであることが、各ループとCPUの接続関係を示す場合と異なる。第14図は各ループにMasterモードのCSTを設定処理中か否かを示すものであり、同時に2つのCSTをMasterモードにすることがないようにインターロックをとるために用いられる。なお、これらのテーブルを参

照するときは、複数CPUから同時に参照されないようインターロックをとる必要があるのでこのテーブルをCPU間共通メモリであるGM4に記した場合は、TEST and SET命令等を用いてインターロックをとる。

次に、CSTの側からの報告によるCSTの構成制御について、第15図、第16図に示す。なお、第11図、第12図、第16図で、他CPUにMasterモードを設定するよう連絡を出すとき、連絡相手を決める方法は、たとえば第13図のループとCSTの接続情報から、ループに接続されているCSTを探し、該CSTを接続しているCPUのうち、切り離し対象となつているCPU、現に停止しているCPUを除いたものから1つを任意に選び出すことによつて行なえる。これは第1図の共通信号線19を用いて行う。なお、この連絡は前述のように、1つのCPUを選び出してから発行してもよいが、全CPUに連絡を出し、連絡を受け取つたCPUが早いもの勝ちで、そのCPUに接続されているCSTをMasterモード

特開昭59- 62968 (8)

とするようにしてもよい。但しその場合、現に Master モードであつたCSTを接続している CPUはこの連絡を無視しなくてはならない。

〔発明の効果〕

このように、本発明によれば、全CSTの状態を共通メモリを用いて各計算機が一括管理して、各CSTの構成制御を行なうので、各CSTのモードの競合、例えば共通バスを一括管理するCSTが複数個発生し、共通バスの管理不能状態を招くという不都合がなくなり、確実なCSTの構成制御を行うことによりマルチ計算機システムの信頼性を向上できる。

図面の簡単な説明

第1図は本発明が適用されるマルチ計算機システムの一実施例構成図、第2図はCSTのモードの状態遷移を示す図、第3図はループバスの運転モードの遷移を示す図、第4図は本発明に用いられるCSTの一実施例ブロック図、第5図〜第9図はそれぞれ第4図の動作説明に用いられるフローチャート、第10図はCPUの構成制御に伴う

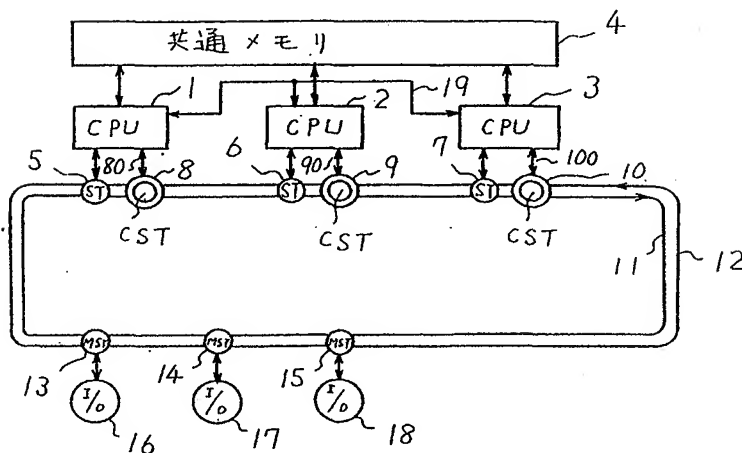
Master モードのCSTの切り換え制御を示す図、第11図はCPUの切り離し指令時の処理を示すフローチャート、第12図はCPUの立上げ時の処理を示すフローチャート、第13図は第12図の動作説明に利用される構成制御情報を示す図、第14図はMaster モードのCSTを設定処理中に用いられる構成制御情報の一例を示す図、第15図はMaster モードのCSTを接続しているCPUの割込み処理を示すフローチャート、第16図はMaster モードでないCSTを接続しているCPUの割込み処理を示すフローチャートである。

1〜3…計算機、4…共通メモリ、8〜10…コントロールステーション(CST)、11、12…ループ状伝送路(共通バス)、80、90、100…接続線。

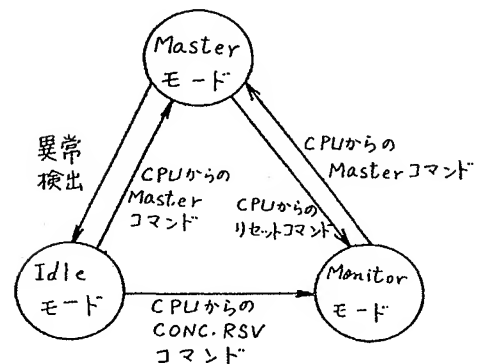
代理人 弁理士 高橋明夫



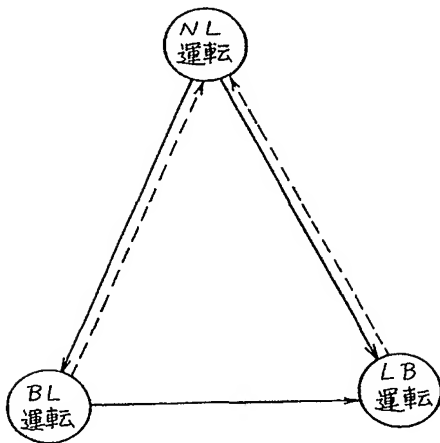
第1図



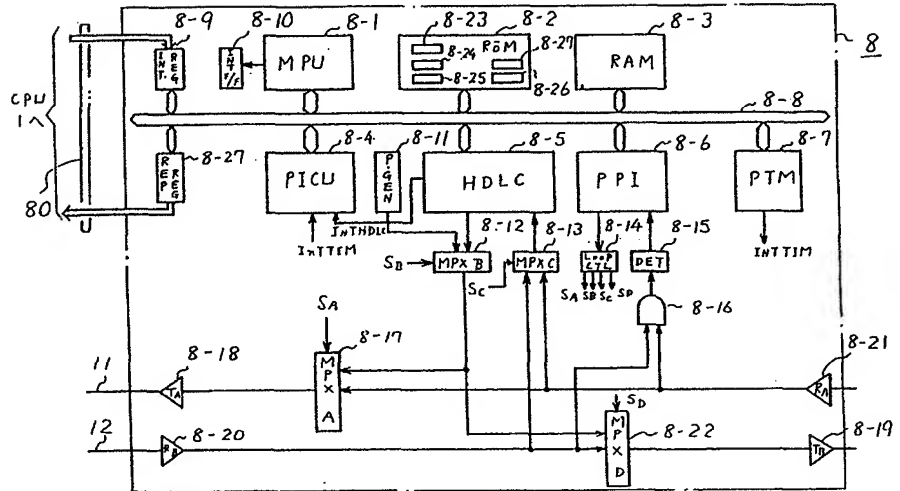
第2図



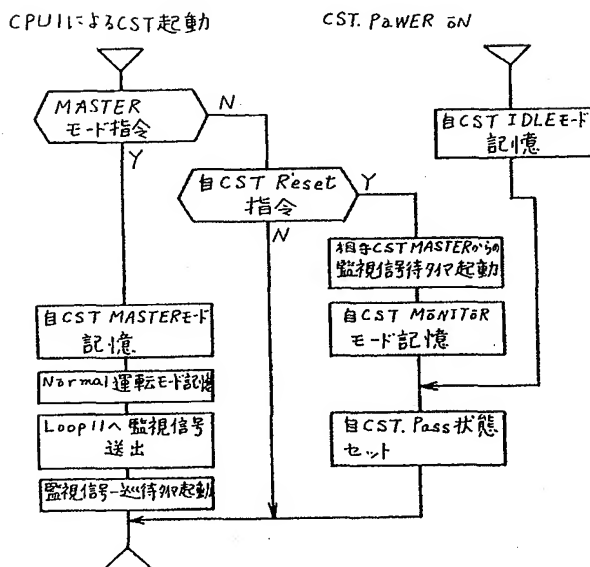
第3図



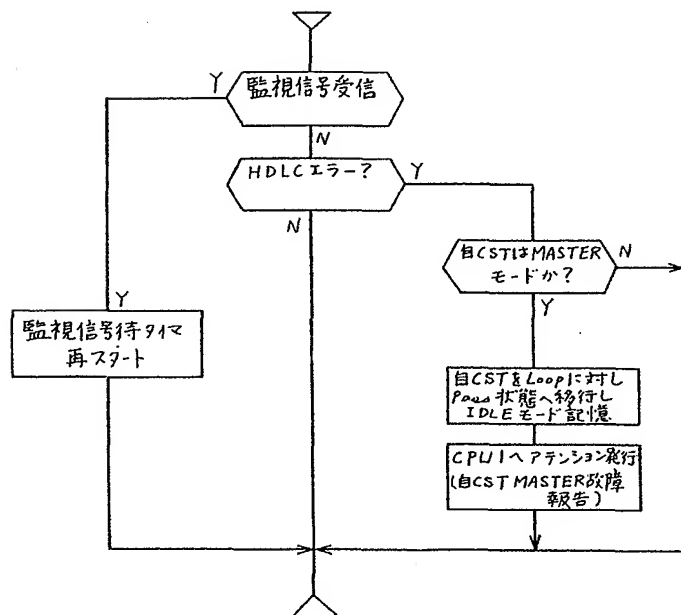
第4図



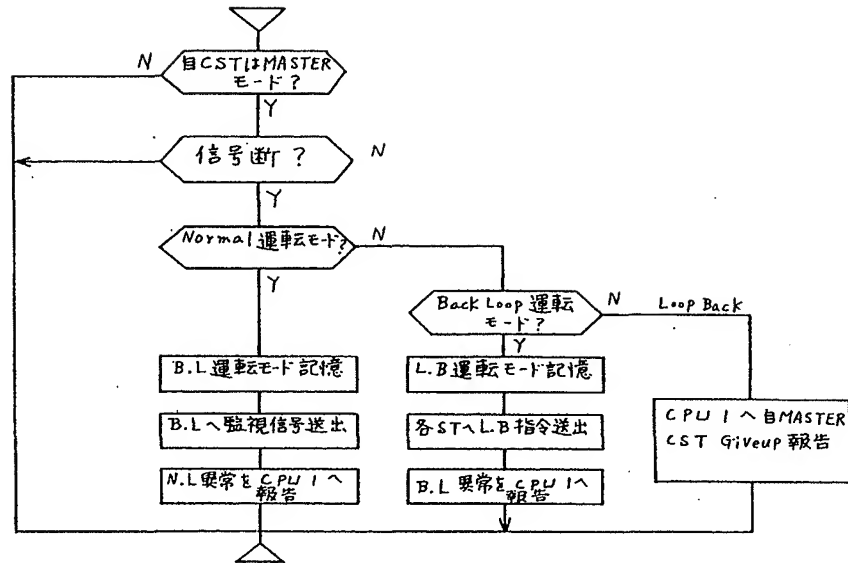
第5図



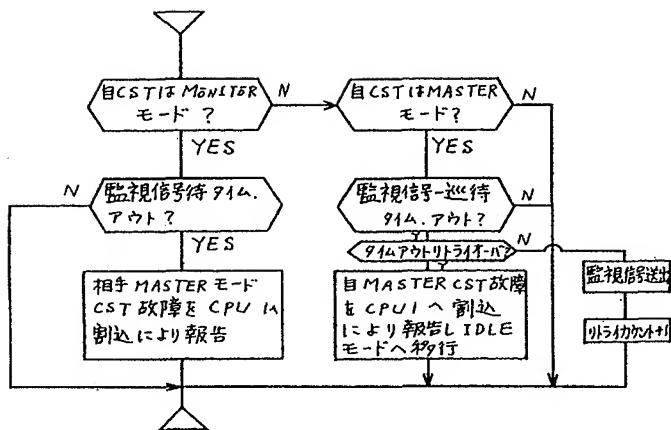
第6図



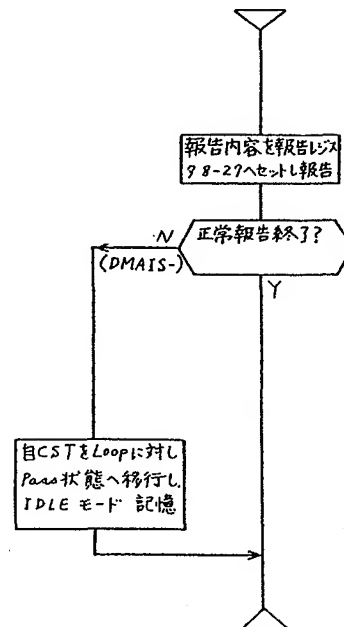
第 7 図



第 8 図



第 9 図

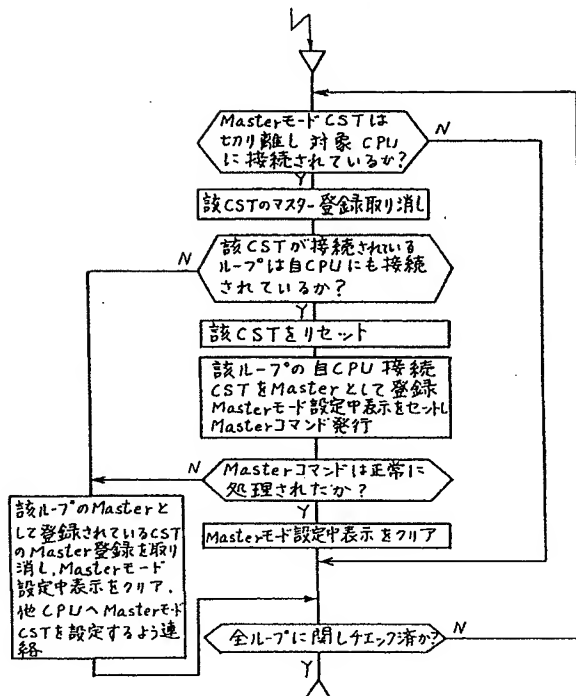


第 10 図

CPUの構成制御	MasterモードCST設定の条件	新しくMasterモードとなるCST
切り離し	MasterモードのCSTが切り離し対象CPUに接続されているとき。	切り離し処理を行なうCPUに接続されてるCST。
切り換え	MasterモードのCSTの変更はしない。	—
立上げ	立上げ処理を行なうCPUに接続されているループにMasterモードのCSTが存在しないとき。	立上げ処理を行なうCPUに接続されているCST

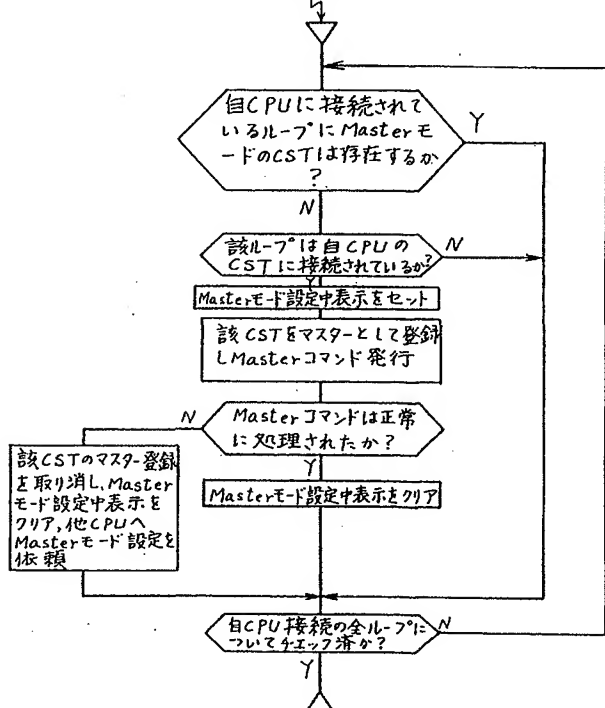
第 11 図

CPUの切り離し指令

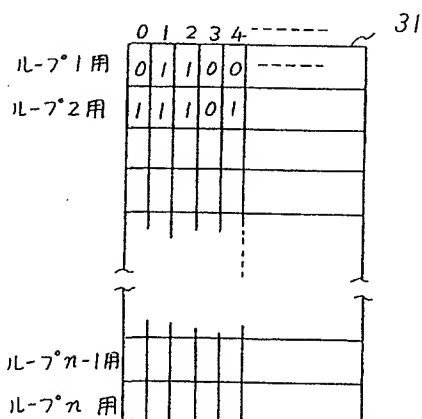


第 12 図

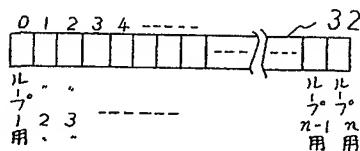
CPUの立上げ



第 13 図

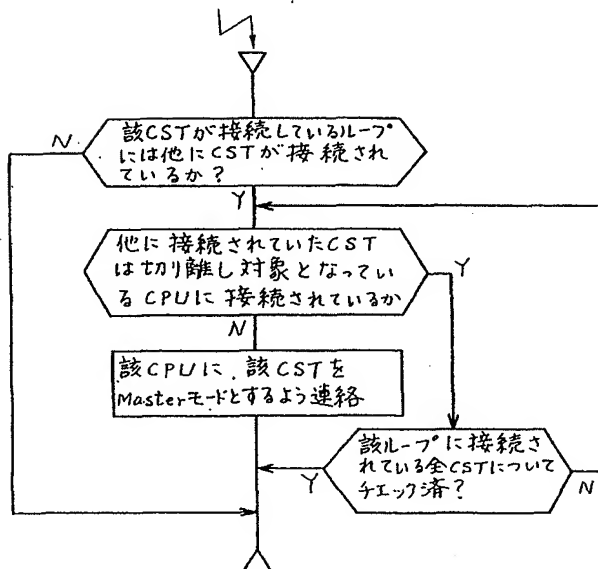


第 14 図



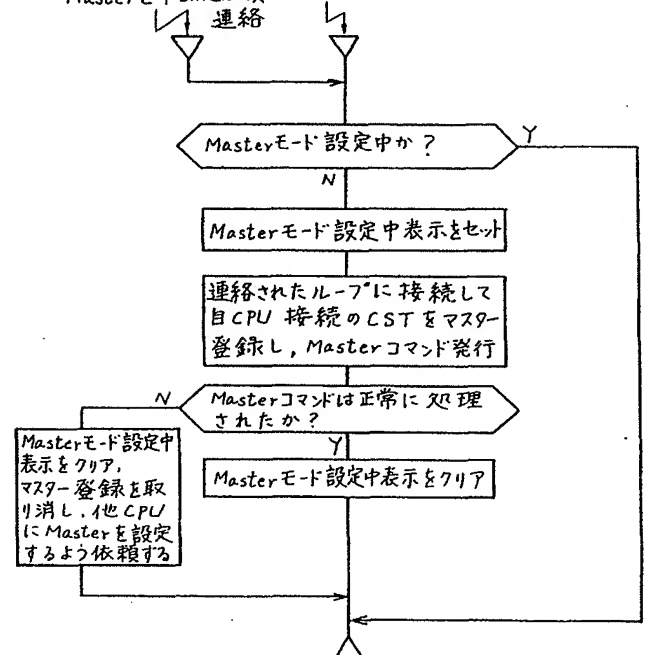
第15図

Masterモード維持不可割り込み



第16図

他CPUからのMasterモード設定依頼 連絡 MasterモードCSTの異常検出割り込み



第1頁の続き

⑫発明者 岡田政和

日立市大みか町5丁目2番1号
株式会社日立製作所大みか工場
内

⑫発明者 末木雅夫

日立市大みか町5丁目2番1号
株式会社日立製作所大みか工場
内

⑫発明者 林慶治郎

日立市大みか町5丁目2番1号
株式会社日立製作所大みか工場
内

⑫発明者 大貫健

日立市大みか町5丁目2番1号
株式会社日立製作所大みか工場
内

⑫発明者 井手寿之

日立市大みか町5丁目2番1号
株式会社日立製作所大みか工場
内

⑫発明者 溝河貞生

日立市大みか町5丁目2番1号
株式会社日立製作所大みか工場
内